Common Project Requirements

MEETS SPECIFICATIONS

App conforms to common standards found in the Android Nanodegree General Project Guidelines

App is written solely in the Java Programming Language

App is written in Java language

App utilizes stable release versions of all libraries, Gradle, and Android Studio.

It uses stable libraries.

Core Platform Development

MEETS SPECIFICATIONS

App integrates a third-party library.

Xkcd API for comics. https://xkcd.com/json.html

App validates all input from servers and users. If data does not exist or is in the wrong format, the app logs this fact and does not crash.

If images of text are not showing it has placeholders. And checks if some field is empty.

App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

Content descriptions are added in all images.

App keeps all strings in a `strings.xml` file and enables RTL layout switching on all layouts.

I keep all strings in strings.xml.

All positions in layout has start or end to support RTL layout and in manifest

android:supportsRtl="true".

MEETS SPECIFICATIONS

App provides a widget to provide relevant information to the user on the home screen.

I've implemented widget if comics has transcript or alternative text, user can click add on transcript button. Widget can be refreshed when you refresh (remove/add) on home screen.

Google Play Services

MEETS SPECIFICATIONS

App integrates two or more Google services. Google service integrations can be a part of Google Play Services or Firebase.

I'm using Admob and Firebase Analitycs.

Each service imported in the `build.gradle` is used in the app.

Admob is added on the main screen of the app and it has test key. Firebase Analitycs has instance in MainActivity.

If `Admob` is used, the app displays test ads. If `Admob` was not used, student meets specifications.

If `Analytics` is used, the app creates only one analytics instance. If `Analytics` was not used, student meets specifications.

Material Design

MEETS SPECIFICATIONS
App theme extends `AppCompat`. MainActivity and main theme extends AppCompat.
App uses an app bar and associated toolbars. It has app bar on every fragment with a back button.
App uses standard and simple transitions between activities. It has navigation through different comics – back and next.

Building

MEETS SPECIFICATIONS
App builds from a clean repository checkout with no additional configuration.
App builds and deploys using the `installRelease` `Gradle` task. It is signed and it has installRelease.
App is equipped with a signing configuration, and the keystore and passwords are included in the repository. Keystore is referred to by a relative path. Keystore config is in the app build gradle.
All app dependencies are managed by `Gradle`.

Data Persistence

MEETS SPECIFICATIONS
App stores data locally either by implementing a ContentProvider OR using Firebase Realtime Database

MEETS SPECIFICATIONS

OR using Room. No third party frameworks nor Persistence Libraries may be used.

It is used room for personal comics collection and SQLite for widget save.

Must implement at least **one** of the three
If it regularly pulls or sends data to/from a web service or API, app updates data in its cache at regular intervals using a SyncAdapter or JobDispacter.

I have implemented JobDispatcher it checks for new comics and sends notification to the user. User can subscribe or remove job.

If Content provider is used, the app uses a Loader to move its data to its views.

If Room is used then LiveData and ViewModel are used when required and no unnecessary calls to the database are made.

I’ve implemented LiveData of request getAllComics.